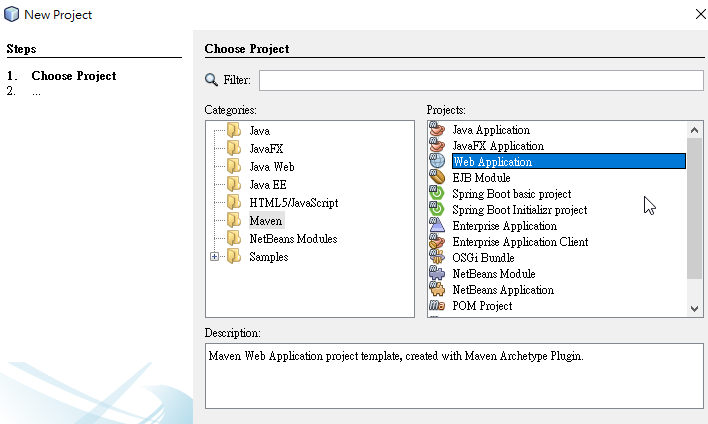
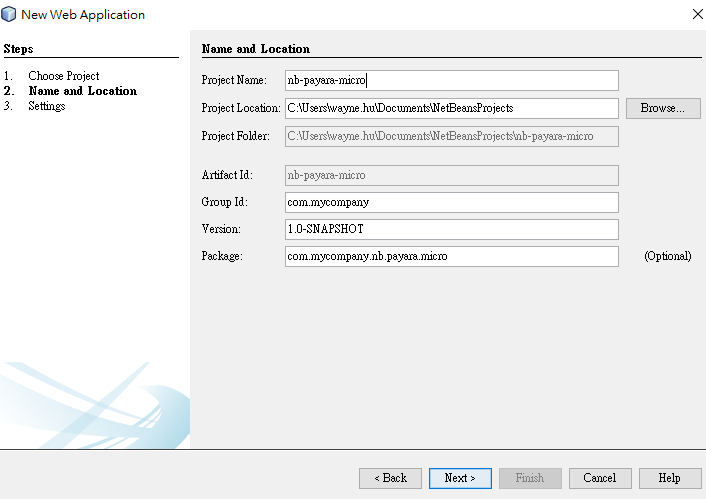
**使用Netbeans & Payara Micro 來開發Micro Service(by Wayne Hu)**

1. 建立新Maven專案,

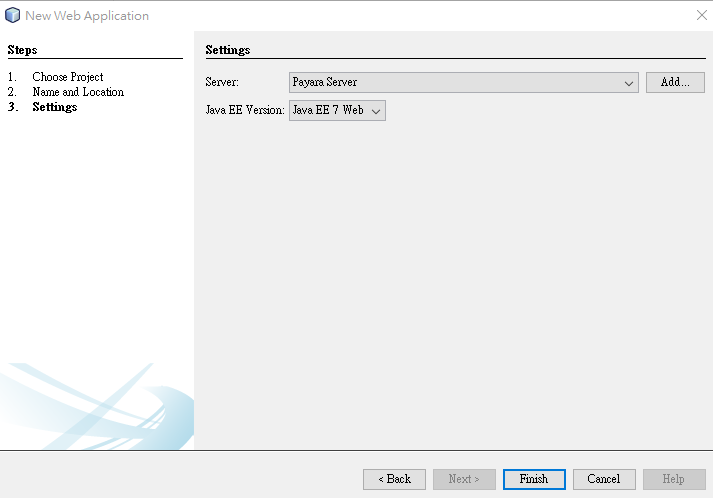
* 於Categories 選擇Maven, Projects 選擇 Web Applications



* 輸入專案名稱

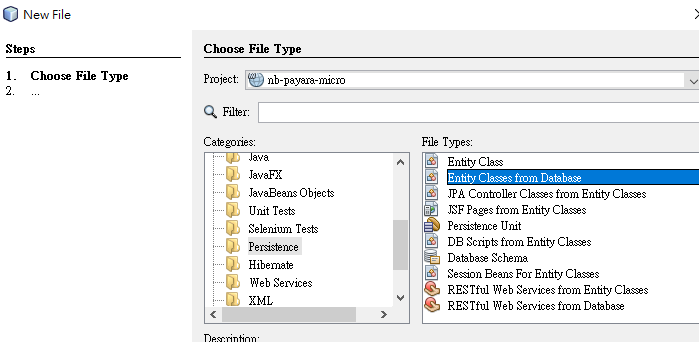


* 於Server 選擇Payara Server.

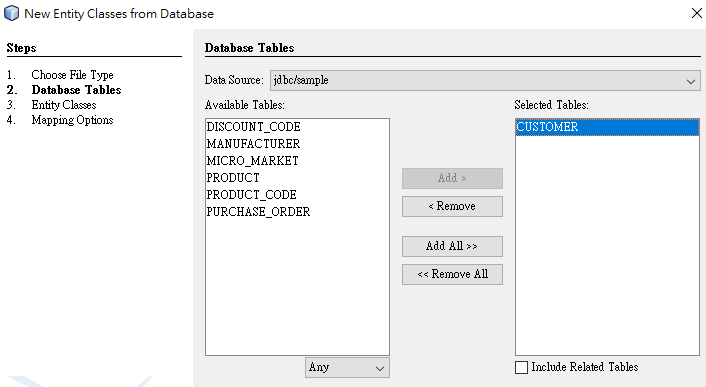


1. 建立JPA Entity

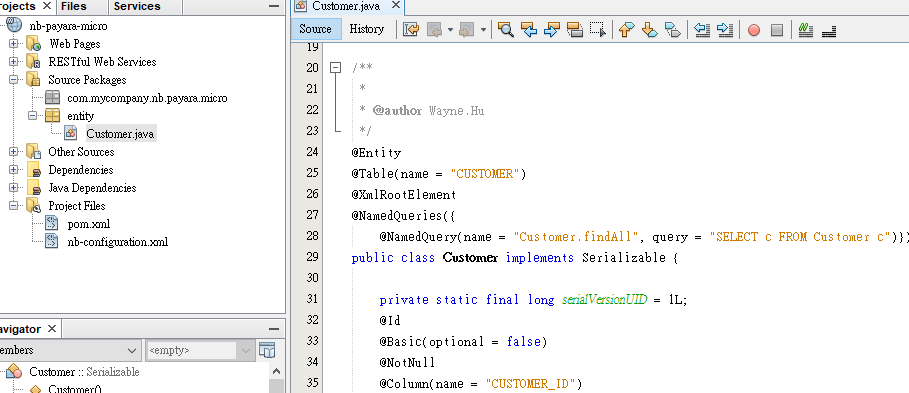
* 選擇New Entity Classes from Database



選擇 Data Source : jdbc/sample, 挑選Customer Table

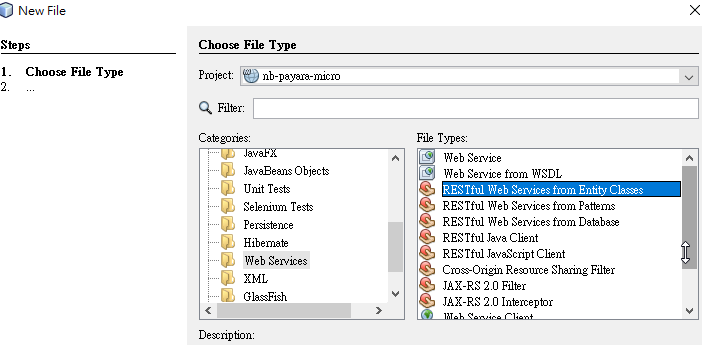


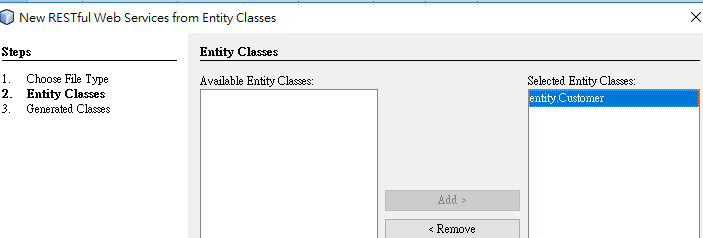
* 建立Entity Customer.java



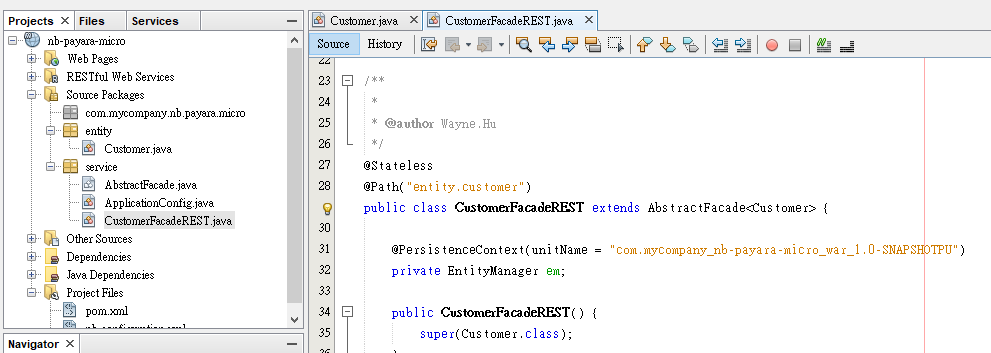
1. 建立Restful Web Service

* 於Categories: Web Services , 選擇建立Restful Web Services from Entity Classes, 並選擇Customer Entity Class.



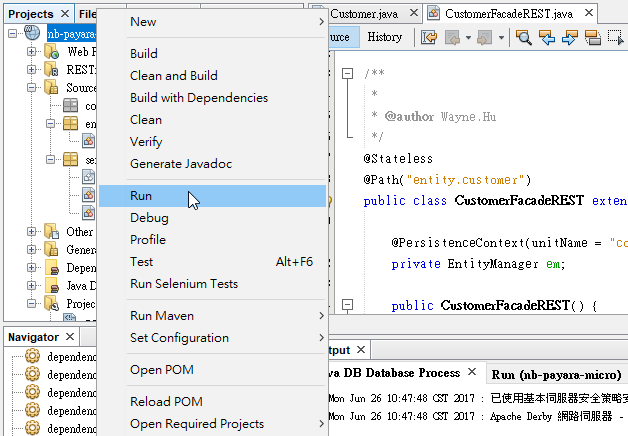


* 成功建立CustomerFacadeREST.java



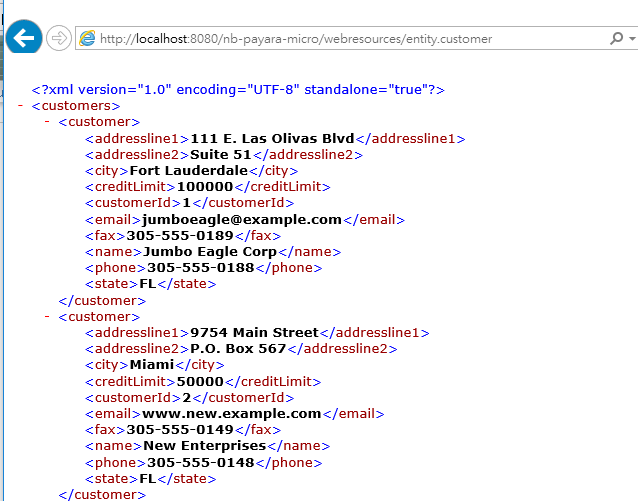
1. 測試Restful Web Service

* 於專案選擇Run



* 並於瀏覽器輸入 <http://localhost:8080/nb-payara-micro/webresources/entity.customer>

得到web service 回傳結果



1. 將專案改成Micro services (Payara Micro) 方式執行

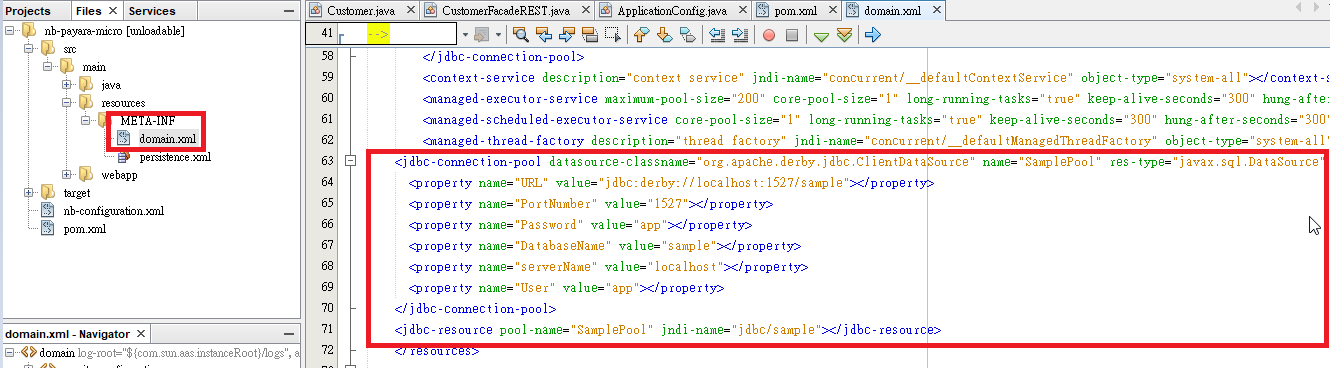
* 於pom.xml 加上payara micro 相關plugin

參考: <https://payara.gitbooks.io/payara-server/documentation/payara-micro/maven/maven.html>

|  |
| --- |
| <dependency>  <groupId>fish.payara.extras</groupId>  <artifactId>payara-micro</artifactId>  <version>4.1.2.172</version>  </dependency> |
| <execution>  <id>copy-payara-micro</id>  <phase>package</phase>  <goals>  <goal>copy</goal>  </goals>  <configuration> <outputDirectory>${project.build.directory}</outputDirectory>  <stripVersion>true</stripVersion>  <silent>true</silent>  <artifactItems>  <artifactItem>  <groupId>fish.payara.extras</groupId>  <artifactId>payara-micro</artifactId>  <type>jar</type>  </artifactItem>  </artifactItems>  </configuration>  </execution> |
| <plugin>  <groupId>org.codehaus.mojo</groupId>  <artifactId>exec-maven-plugin</artifactId>  <version>1.6.0</version>  <executions>  <execution>  <goals>  <goal>exec</goal>  </goals>  </execution>  </executions>  <configuration>  <executable>java</executable>  <arguments>  <argument>-jar</argument>  <argument>${project.build.directory}/payara-micro.jar</argument>  <argument>--deploy</argument>  <argument>${project.build.directory}/${project.build.finalName}.war</argument>  <argument>--domainConfig</argument>  <argument>${project.basedir}/src/main/resources/META-INF/domain.xml</argument>  </arguments>  </configuration>  </plugin> |

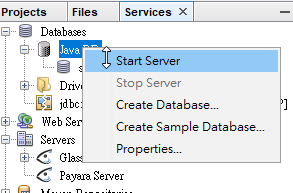
* 因上述執行參數--domainConfig需要一個自己專案的domain.xml , 作法是由Payara Micro 中將domain.xml 抽離出來, 置放於專案目錄中, 然後將前面 Payara Server 中的domain.xml 將 </resources> 上方的jdbc-connction-pool 及 </resources> 複製到專案中的domain.xml

參考文件: <https://stackoverflow.com/questions/32899120/how-do-i-add-jdbc-drivers-and-configure-jdbc-resources-in-payara-micro>

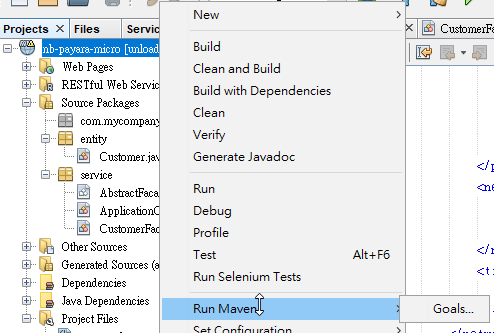


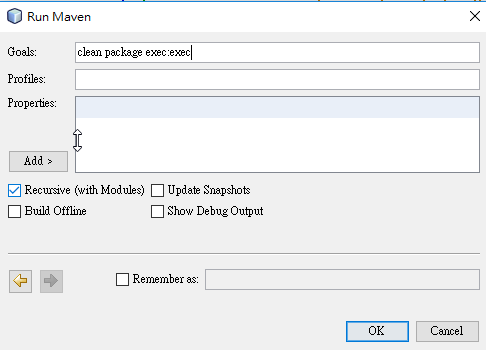
1. 透過Maven 執行Micro service

* 停掉Payara Server 以釋出port #8080, 然後手動啟動 Java DB



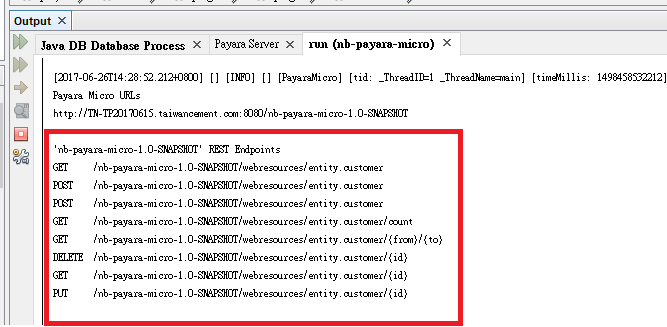
* 於 專案中, 選擇Run Maven, 並輸入Goals : clean package exec:exec



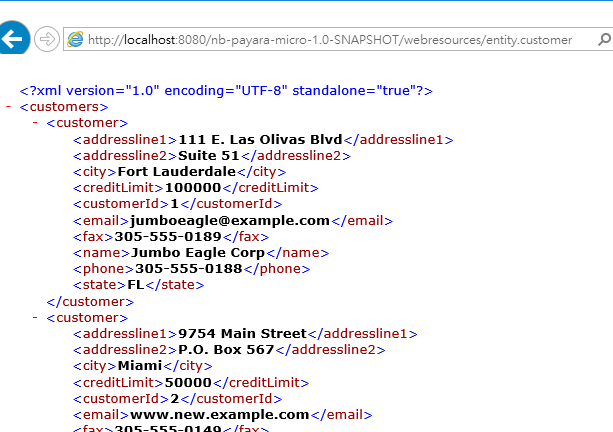


1. 測試結果

* Payara Micro 執行後, Console 會顯示web service URL (注意: URL 有加上-1.0-SNAPSHOT)



* 於瀏覽器輸入URL: <http://localhost:8080/nb-payara-micro-1.0-SNAPSHOT/webresources/entity.customer>



* 在下次執行前, 需於Netbean/Run 中停掉執行中的Micro service

